A Phenomic Algorithm for Inference of Gene Networks Using S-Systems and Memetic Search

Rio G.L. D'Souza[1], K. Chandra Sekaran[2], and A. Kandasamy[2]

[1] St Joseph Engineering College, Mangalore, India
[2] National Institute of Technology Karnataka, Surathkal, Mangalore, India
{Rio,kchnitk}@ieee.org, kandy@nitk.ac.in

Abstract. In recent years, evolutionary methods have seen unprecedented success in elucidation of gene networks, especially from microarray data. We have implemented the Phenomic Algorithm which is an evolutionary method for inference of gene networks based on population dynamics. We have used S-systems to model gene interactions and applied memetic search to fine tune the parameters of the inferred networks. We have tested the novel algorithm on artificial gene expression datasets obtained from simulated gene networks. We have also compared the results to those obtained from two other similar algorithms. Results showed that the new method, which we call as Phenomic Algorithm with Memetic Search (PAMS), is an effective method for inference of gene networks.

Keywords: Microarray data analysis, Gene networks, Evolutionary algorithms, S-systems, Memetic search, Phenomic algorithms.

1 Introduction

Ever since the advent microarray technology scientists have been able to study thousands of genes at a time, and this has helped them to analyze the relationships between them. Most microarray experiments result in large datasets which need to be analyzed in order to understand the underlying relationships. There is vast potential for methods that can yield useful patterns from such large datasets without compromising the dimensionality [1]. Gene networks represent relationships between genes, based on observations of how the expression level of each gene affects the expression levels of the others [2]. Several researchers have used evolutionary methods [3] to analyze the relationships between thousands of genes. The Phenomic Algorithm [4], [5] is an approach based on population dynamics. We have implemented the proposed algorithm and validated it on artificial gene network datasets.

The rest of this paper is organized as follows: In Section 2, we provide a review of similar work done by others. We introduce the models and also the basis of the methods that we employ in Section 3; and in Section 4 we discuss the results of our experiments. This is followed by Section 5 which concludes the paper.

J. Suzuki and T. Nakano (Eds.): BIONETICS 2010, LNICST 87, pp. 229–237, 2012.

2 Related Work

The Inference of gene networks from the ever-growing mass of microarray data has become an important research activity in Systems Biology. Among the initial attempts, Somogyi et al. [6] developed a simple method which inferred Boolean networks. Several gene network reconstruction algorithms have been studied by Akutsu et al. [7] and D'haeseleer et al. [8]. While some of these methods infer only qualitative relationships between genes, others which infer quantitative relationships are limited by the scale of networks that they can deduce.

Reliable inference of gene networks is dependent on how closely the chosen model represents the real gene networks. One such model, which is nonlinear and dynamic is the S-System proposed by Savageau [9]. Several researchers [10] have used this model to reverse engineer gene networks. Recently, Ahmed, Song and Xing [11] have used a variant of S-System to construct graphical models for inferring time-varying gene regulatory networks. Most problems in this field can be viewed as some type of optimization and multiobjective evolutionary algorithms (MOEAs) have found remarkable success in reconstructing gene networks from expression data [12].

3 Models and Methods

3.1 S-System Model of Gene Networks

To establish that a change in the expression of gene B was caused by a change in the expression of gene A, it is necessary to show that a dependency exists between the two genes, whereby gene B is dependent on gene A. The Power-law formalism called S-System, which was proposed by Savageau [9] is a nonlinear and dynamic model which we used to capture the relationships between genes.

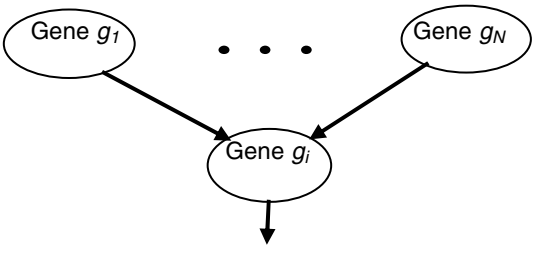

Fig. 1. Generalized gene network model

The behaviour of a cell can be abstracted by a gene regulatory network of N genes and other intermediate gene-products. Each gene g_i produces a certain amount of RNA x_i whenever it expresses. This causes a change in the concentration of this RNA over a time-period. This situation, shown in Fig. 1, can be represented in equation (1):

$$x(t+1)=h(x(t)), \qquad where\ x(t)=(x_1, x_2, \ldots, x_N)\ . \tag{1}$$

The S-system model for this gene network can be described by a set of nonlinear differential equations in equation (2):

$$\frac{dx_i(t)}{dt} = \alpha_i \prod_{j=1}^{N} x_j(t)^{G_{i,j}} - \beta_i \prod_{j=1}^{N} x_j(t)^{H_{i,j}} \quad . \tag{2}$$

Here $G_{i,j}$ and $H_{i,j}$ are kinetic exponents and α_i and β_i are positive rate constants. Ignoring genes that do not influence gene g_i, applying the rule of finite differences and rearranging equation (2), the calculated value of gene expression level for gene g_i, $x_{i,cal,t+1}$ can be written as in equation (3):

$$x_{i,cal,t+1} = x_i(t+1) = x_i(t) + \left\{ \alpha_i \prod_{j \in S_i} x_j(t)^{G_{i,j}} - \beta_i \prod_{j \in S_i} x_j(t)^{H_{i,j}} \right\} \Delta t \quad . \tag{3}$$

After each evaluation of $x_{i,cal,t+1}$, the error at that time-step between the calculated and observed gene expression level is given in equation (4) as:

$$e_{i,t} = x_{i,cal,t} - x_{i,\exp,t} \quad . \tag{4}$$

The link that is being verified is retained if the error is less than E (which is the maximum error allowed at that point in the inference algorithm), otherwise the link is not retained. The kinetic exponents and rate constants are optimized separately through a memetic search mechanism.

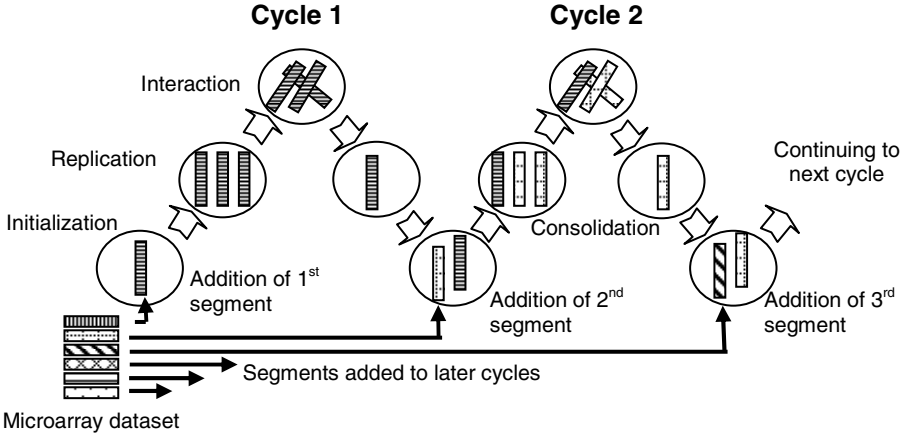

Fig. 2. First two cycles of processing in the phenomic algorithm

3.2 The Phenomic Approach

In order to elucidate gene-to-gene relationships from expression measurements taken from microarray data, the expression patterns of thousands of pairs of genes need to be compared. Carrying out this huge number of comparisons in a coordinated manner,

such that all relationships of interest are discovered, is a challenging task. Modeled carefully gene expression data could be used to characterize the relationship between the phenotype and the genotype.

In this approach, as seen in Fig. 2, the individuals interact and it is during these interactions that the relationships between genes are captured. A link is set up between two genes that are causally related. After a predetermined number of randomized interaction cycles, the population is consolidated to remove replicated individuals. The links captured by dying individuals are carried over to the survivors.

3.3 Memetic Search

In equation (3), the set of kinetic rate constants and exponents $\{\alpha_i , \beta_i , G_{i,j} , H_{i,j}\}$ determine the change in the expression level of gene g_i at any time t, due to the change in expression levels of all genes at that time. The parameters associated with a gene network are optimized by the memetic search procedure, as given below:

The memetic search procedure.

```
Procedure MemeticSearch(geneNetwork, kineticString)
begin
     t := 0;
     P(t) := initPop(kineticString);
     similarity := 0.0;
     while (similarity < 0.6)
          P'(t) := generateOffspringPop(P(t));
          P(t+1) := retainBetterString(P(t), P'(t));
          similarity := evaluateSimilarity(P(t+1));
          t := t+1;
     endwhile
     kineticString := findBest(P(t));
end
```

3.4 The Phenomic Algorithm with Memetic Search (PAMS)

Like most evolutionary algorithms, the phenomic algorithm with memetic search starts with a population of n individuals each of which embeds the expression profile of one gene taken from the microarray data. The algorithm thereafter goes into the evolutionary cycle which starts with the first generation of phenotypic processing. During phenotypic processing the triplets of individuals (or genes) are allowed to meet in the interaction phase. The possibility of causal relationship between them is verified by using equation (4). The memetic search procedure is used at this stage to fine-tune the network parameters.

The process is repeated starting with the interaction phase, until all microarray data segments are processed. The gene networks at this stage are output as the optimal

networks that represent the given microarray data. The pseudo-code of PAMS is given below:

The phenomic algorithm with memetic search (PAMS) and its main functions.

```
  Algorithm PAMS
  begin
      t := 0;
      segments[] := divideMicroarrayData();
      P(t) := replicateSeg(segments[t]);
      while (t < n)
            P'(t) := interactPop(P(t));
            P"(t) := consolidatePop(P'(t));
            P(t+1) := P"(t) + replicateSeg(segments[t+1]);
            t := t+1;
      endwhile
      geneLinks := readLinks(P(t));
      displayNetworks(geneLinks);
end
```

In the next section, PAMS is validated and compared with other extant multiobjective algorithms for inferring gene networks.

4 Results and Discussion

The validation of PAMS was done by using datasets derived from artificial gene networks developed by researchers at the Virginia Bioinformatics Institute and downloadable from their website [13]. Two datasets were selected: one is called Century and the other is called Jumbo. A typical gene network inferred when running PAMS on the Century dataset is shown in Fig. 3.

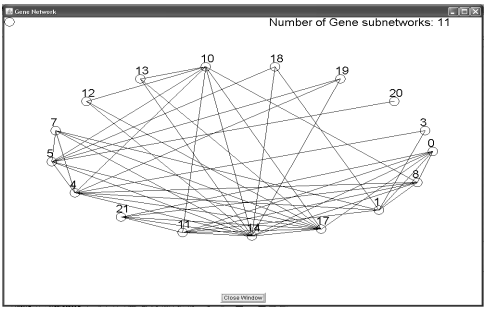

Fig. 3. A gene network inferred by PAMS when E = 0.05

We now define and use three metrics as the basis for comparison of the three methods. The first metric, NOL, is defined as in equation (5). The second metric, SWSF, measures the closeness of the inferred networks to small-world networks and is defined in equation (6).

$$NOL = \sum_{i=1}^{N} \sum_{j=1}^{N} l_{ij} \quad . \tag{5}$$

$$SWSF = \frac{1}{C} \sum_{k=1}^{C} k \, n_k \quad . \tag{6}$$

Here, $l_{ij} = 1$ if gene g_i is linked to gene g_j, else $l_{ij} = 0$ (taken from the adjacency matrix of the network), N is the total number of genes in the target network, n_k is the number of nodes with out-degree of k, and C is the maximum degree of the network.

The third metric, GED, is the minimal number of edit operations (edge insertions/edge removals) that transform one graph into another one [14]. In our case, we use the formula in equation (7) to calculate this distance.

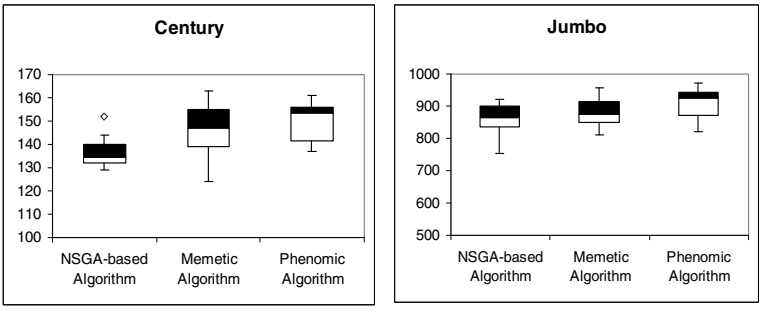

Fig. 4. Boxplots of NOL obtained with Century and Jumbo gene network datasets

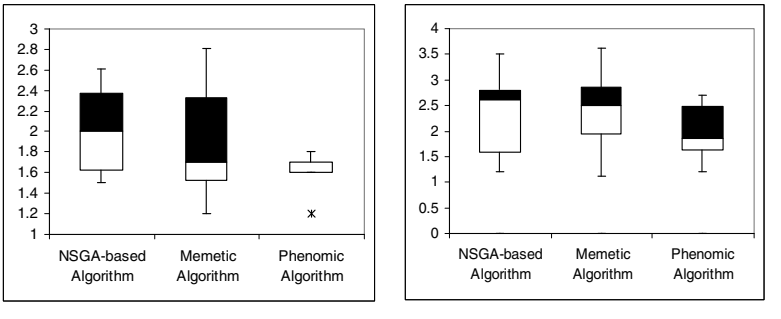

Fig. 5. Boxplots of SWSF obtained with Century and Jumbo gene network datasets

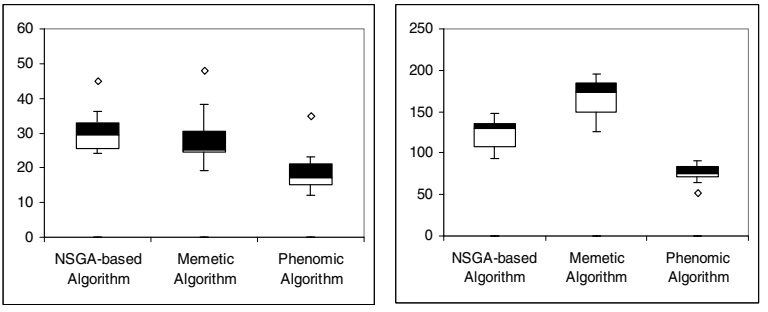

Fig. 6. Boxplots of GED obtained with Century and Jumbo gene network datasets

$$GED = \sum_{i=1}^{N}\sum_{j=1}^{N} D_{ij} \qquad \begin{cases} D_{ij} = 1, if\ l_{ij} \neq m_{ij} \\ D_{ij} = 0, if\ l_{ij} = m_{ij} \end{cases} . \tag{7}$$

Here D_{ij} is the distance between corresponding nodes of the two networks that are being compared; l_{ij} and m_{ij} are elements of the adjacency matrices of the inferred gene network and source artificial gene network, respectively.

The phenomic algorithm (PAMS) is compared with two other evolutionary algorithms that are used for inferring gene networks. The first of these algorithms is based on Non-dominated Sorting Genetic Algorithm (NSGA-based) which was proposed by Deb et al. [15], combined with a gene network inference algorithm as implemented by Spieth et al. [16]. The second algorithm is a Memetic Algorithm developed by Spieth et al. [17]. For further details readers are referred to the original papers. The descriptive statistics of the three metrics over ten runs of each of the above methods are shown as boxplots in Fig. 4, Fig. 5 and Fig. 6.

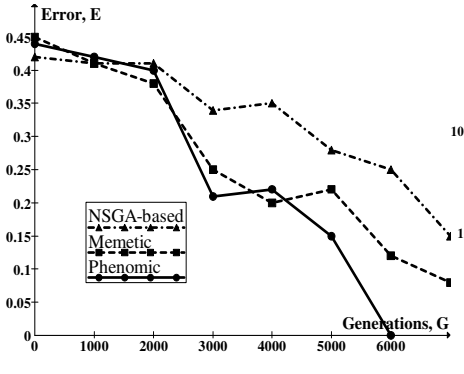

Fig. 7. Variation of error observed against generations

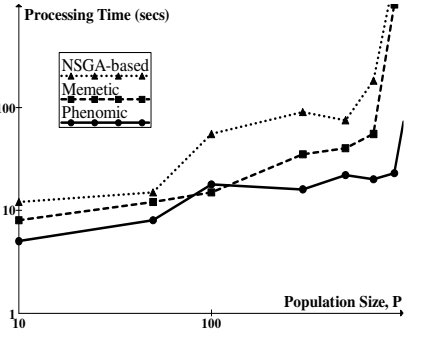

Fig. 8. Variation of processing time against population size

The boxplots in Fig. 4, Fig. 5 and Fig. 6 show that PAMS infers qualitatively better networks than both the NSGA-based and the Memetic algorithm. This is achieved without compromising on the runtime performance as shown in Fig. 7 and Fig 8. In terms of error E, as shown in Fig. 7, the phenomic algorithm achieves zero error after about 6000 generations. All values plotted are the average of 10 runs. The processing time taken by PAMS for processing even large populations (approaching 1000 individuals), is less than a minute.

5 Conclusion

The problem of inferring gene networks is getting increased attention from researchers worldwide and many new methods are introduced each year. Multiobjective evolutionary algorithms have been yielding consistently good results [12] and the phenomic algorithm adds to this successful trend. The benefits of local search through memetic mechanisms are also evident in the results described earlier.

In future work, it is planned to apply PAMS to natural gene expression datasets. The robustness of the novel method to noise, which is a frequent spoiler in natural datasets, will be under scrutiny in such experiments.

References

1. Schulze, A., Downward, J.: Navigating Gene Expression Using Microarrays – A Technology Review. Nature Cell Biology 3, E190–E195 (2001)
2. Soinov, L.A., Krestyaninova, M.A., Brazma, A.: Towards Reconstruction of Gene Networks from Expression Data by Supervised Learning. Genome Biology 4(1), R6 (2003)
3. D'haeseleer, P., Liang, S., Somogyi, R.: Gene Expression Analysis and Genetic Network Modelling: Tutorial. In: Pacific Symposium on Biocomputing (1999)
4. D'Souza, R.G.L., Chandra Sekaran, K., Kandasamy, A.: A Phenomic Algorithm for Reconstruction of Gene Networks. In: IV International Conference on Computational Intelligence and Cognitive Informatics, CICI 2007, pp. 53–58. WASET, Venice (2007)
5. D'Souza, R.G.L., Chandra Sekaran, K., Kandasamy, A.: Reconstruction of Gene Networks using Phenomic Algorithms. Intl. Jour. of Artificial Intell. and Appl. 1(2) (2010)
6. Somogyi, R., Fuhrman, S., Askenazi, M., Wuensche, A.: The Gene Expression Matrix: Towards the Extraction of Genetic Network Architectures. In: Proc. of Second World Cong. of Nonlinear Analysts (WCNA 1996), vol. 30(3), pp. 1815–1824 (1997)
7. Akutsu, T., Miyano, S., Kuhara, S.: Identification of Genetic Networks from a Small Number of Gene Expression Patterns under the Boolean Network Model. In: Pacific Symp. on Biocomputing, vol. 4, pp. 17–28 (1999)
8. D'haeseleer, P., Liang, S., Somogyi, R.: Genetic Network Inference: From Co-Expression Clustering to Reverse Engineering. Bioinformatics 16(8), 707–726 (2000)
9. Savageau, M.A.: Power-law Formalism: A Canonical Nonlinear Approach to Modelling and Analysis. In: Proc. of the World Congress of Nonlinear Analysts 1992, pp. 3323–3334 (1995)

10. Spieth, C., Streichert, F., Speer, N., Zell, A.: Optimizing Topology and Parameters of Gene Regulatory Network Models from Time-Series Experiments. In: Deb, K., Tari, Z. (eds.) GECCO 2004. LNCS, vol. 3102, pp. 461–470. Springer, Heidelberg (2004)
11. Ahmed, A., Song, L., Xing, E.: Time-Varying Networks: Reconstructing Temporally Rewiring Genetic Interactions during the Life Cycle of Drosophila melanogaster. CMU-MLD Technical Report CMU-ML-08-118 (2008)
12. Van Veldhuizen, D.A., Lamont, G.B.: Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art. Evolutionary Computation 8(2), 125–147 (2000)
13. A Collection of Artificial Gene Networks (2010), `http://www.comp-sys-bio.org/AGN/data.html` (accessed January 20, 2010)
14. Supper, J., Fröhlich, H., Spieth, C., Dräger, A., Zell, A.: Inferring Gene Regulatory Networks by Machine Learning Methods. In: APBC 2007, pp. 247–256 (2007)
15. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II. IEEE Trans. of Evol. Comp. 6(2), 182–197 (2002)
16. Spieth, C., Streichert, F., Speer, N., Zell, A.: Multi-objective Model Optimization for Inferring Gene Regulatory Networks. In: Coello Coello, C.A., Hernández Aguirre, A., Zitzler, E. (eds.) EMO 2005. LNCS, vol. 3410, pp. 607–620. Springer, Heidelberg (2005)
17. Spieth, C., Streichert, F., Speer, N., Zell, A.: A Memetic Inference Method for Gene Regulatory Networks Based on S-Systems. In: Congress on Evol. Comp (CEC 2004), Proc. Part I, pp. 152–157. IEEE Press (2004)